523

NPS-OR-94-005

NAVAL POSTGRADUATE SCHOOL

Monterey, California

UNITED STATES NAVY
NAVAL POSTGRADUATE SCHOOL

THE NPS PLATFORM FOUNDATION

Michael P. Bailey

February 1994

Prepared for:
Naval Postgraduate School
Monterey, CA 93943

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943-5000

This report was prepared for the Naval Postgraduate School.

Reproduction of all or part of this report is authorized.

This report was prepared by:

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE 15 Feb 1994	3. REPORT TYPE AND DATES COVERED Technical

4. TITLE AND SUBTITLE The NPS Platform Foundation	5. FUNDING NUMBERS
6. AUTHOR(S) Michael P. Bailey	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943	8. PERFORMING ORGANIZATION REPORT NUMBER NPS-OR-94-005
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	10. SPONSORING / MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES	
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

There are many well-adapted commercial simulation tools for specific problem domains. Many vendors concentrate on manufacturing, communications, and computer networks applications. The NPS Platform Foundation is a tool for modeling military platform engagements, and will support construction of a wide variety of models where platforms, sensors, humans, tactics, and information flow are important.

Analysts (e.g. NPS thesis students) can use the Foundation's generic platform to configure or tailor objects to meet specific project needs by adding data to the performance database, by adding a layer of tactical methods, or by refining platform motion and sensor performance methods.

14. SUBJECT TERMS Object-oriented simulation modeling	15. NUMBER OF PAGES 21
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

The NPS Platform Foundation

Michael P. Bailey

Department of Operations Research

Naval Postgraduate School

Monterey, California 93943-5000

mike@uwhiz.or.nps.navy.mil

February 15, 1994

Abstract

There are many well-adapted commercial simulation tools for specific problem domains. Many vendors concentrate on manufacturing, communications, and computer networks applications. The NPS Platform Foundation is a tool for modeling military platform engagements, and will support construction of a wide variety of models where platforms, sensors, humans, tactics, and information flow are important.

Analysts (eg. NPS thesis students) can use the Foundation's generic platform to configure or tailor objects to meet specific project needs by adding data to the performance database, by adding a layer of tactical methods, or by refining platform motion and sensor performance methods.

1 Introduction

Many simulation application domains are composed of identifiable key components – communications analysis involves packets, transmitters, receivers, processors, and protocols; assembly lines are composed of workstations, partially assembled pieces, and conveyers. There exist specialized simulation software packages designed for each of these domains, allowing analysts to generate sophistocated models quickly. Military operations analysts work in a domain where the key components are platforms, weapons, sensors, and tactics. However, at this writing there is no comercial, off-the-shelf simulation software product which is designed to model the military tactical domain.

Lower level tools such as third-generation computer languages or general-purpose simulation languages have been used to build military simulation models. The result has been that military simulations have been expensive to build, and the focus of the military operations analyst has been directed toward software development rather than model analysis.

The Naval Postgraduate School Platform Foundation is a collection of objects which provide the functionalities required to model situations where platforms interact using sensors, weapons, and tactics. The focus of platform engagement scenarios typically involve questions such as:

- Who can see whom?
- Where does platform A think platform B is?
- How well does an evasion or pursuit tactic work?
- How can improved detection equipment, improved movement performance, or increased weapon effectiveness change a typical engagement's outcome?
- How can tactics be adapted to exploit improved equipment?
- How does training effect an engagement's outcome?

Each of these questions have answers which evolve in time and space, so simulation models are often used to produce answers.

Text-based or statistical characterizations of the behaviors of platforms in a scenario are often not useful because of the complexity and time-spatial nature of a scenario. An animated map display can show an analyst what is happenning and when. Accompanied with text-based output of each platform's state, an animation of the scenario is critical to the understanding of the dynamics of a multiplatform engagement.

Animation, while highly desirable, is also hard to manage, expensive to develop, and difficult to port between computer platforms. For these reasons, it was essential that the Platform Foundation include automatic, portable animation capability. With no added effort, platforms are represented as animated icons, with an animated range ring associated with each active, mounted sensor, moving around on a Foundation-generated, zoomable map.

The Platform Foundation is written in the object-oriented language MODSIM [7]. We chose the name because the objects form a foundation on which specific simulations can be constructed, and because the development of the Foundation coincided with the death of the famed scientist and author Isaac Azimoz. The Foundation was designed as part a course in simulation taught at the Naval Postgraduate School, and was designed, implemented, and tested in a two-month period. The Foundation is, at this writing, over 17,000 lines of MODSIM, and 3,000 lines of C.

2 The Line

Platform Foundation objects (`PlatformObjs`) possess a set of generic capabilities which provide all that is needed to build simple simulation models. Simply by changing a set of data files, an analyst can mount weapons and sensors on platforms, and make the platforms execute complex maneuver sequences. All the action is displayed on a geographic situation display. Finally, the platforms each issue periodic situation updates using the Distributed Interactive Simulation protocol. All of these capabilities work in a more or less automatic way. Figure 1 shows these capabilities as those generic, lying *below the line.*

The *line* is the division between generic Foundation capabilities and functions which are programmed for a specific application. In terms of our software design, the objects below the line are the `PlatformObj`, the `SensorObj`, and the `WeaponObj`, and all of the scenario management software. The intent is that objects below the line are incorporated in special-purpose, application-specific objects like radars, ships, aircraft, and missles. These *above-the-line* objects can use *below-the-line* capabilities as primitive elements in their tactical methodology.

As an example, consider an ASW aircraft. It could have a procedure (MODSIM METHOD)

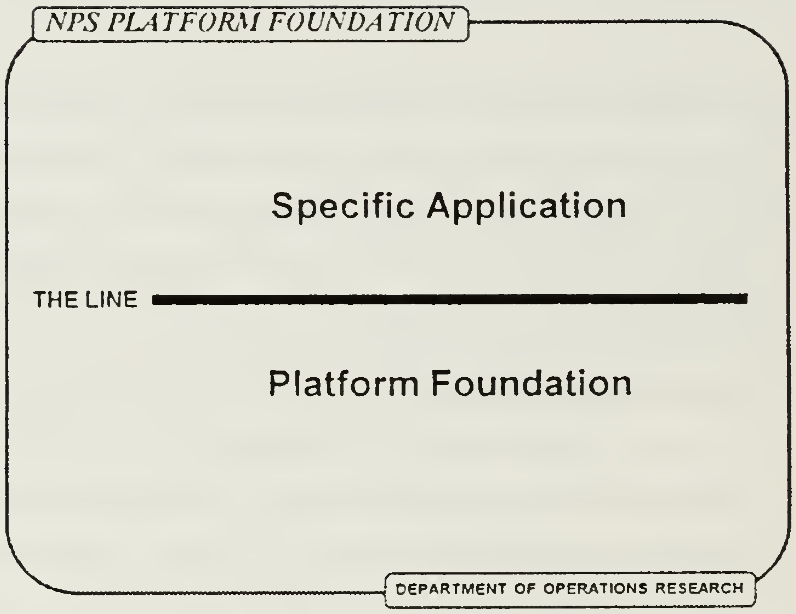

Figure 1: The LINE, dividing generic Platform Foundation capabilities from application-specific additions.

in which it transits to a location, flies a race track pattern as it sews a pattern of sonobuoys, flies to a new stationing location, and loiters in another race track pattern. This METHOD could be accomplished using the primitive `RaceTrack` and `MoveStraightTo` movement METHODs, along with several `ExpendWeapons` (later we explain why sonobuoys are convieniently modeled as weapons).

Figure 2 shows a partial OOSPic [1] object layout, a graphical representation of the relationships of objects in the simulation. The objects above the line are special objects used to construct a fixed-wing ASW simulation (see section 5.1). The lay-out shows the `PlatformObj` being an ancestor of the `SystemManagerObj`. The `SystemManagerObj` owns a `SensorSuite` and a `WeaponSuite` which house sensors and weapons. Above the line, we see specialized weapons, sensors, and platforms designed specifically for the ASW simulation.

`RaceTrack` and `MoveStraightTo` are below-the-line capabilities of the `PlatformObj`. They cause the correct timing, cartographic, navigation, and animation actions to occur automatically. Thus, the analyst interested in fixed-wing ASW aircraft can quickly construct tactical METHODs by exploiting the below-the-line capabilities of the `PlatformObj`, `WeaponObj`, and `SensorObj`.

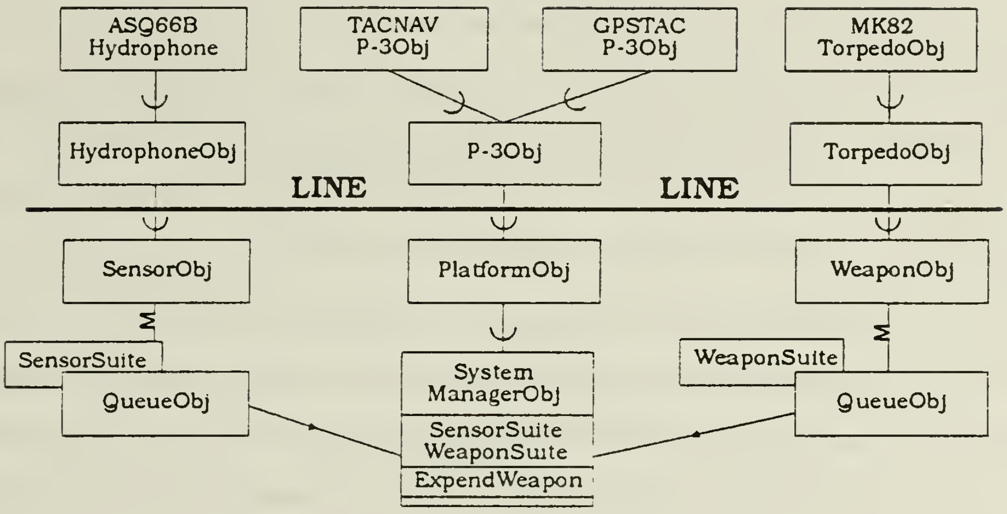

Figure 2: An example of the below- and above-the-line objects. The arrows with semicircles indicate inheritance, the object on the open end is the ancestor. Lines with "M" indicate membership in sets, and the triangles on lines point toward owners of other objects.

3 Foundation Architecture

In this section, we give brief descriptions of the functional components of platforms, sensors, and weapons. It is no accident that these descriptions look like lists of increasingly complex behaviors, as the Platform Foundation is based on a functionally decomposed object-oriented design, [5].

3.1 Platforms

Platforms are things which move through space and time, using sensors and weapons to interact with other platforms. The Platform Foundation allows an analyst to concentrate on tactics and doctrine because it provides key capabilities which every platform needs. These capabilities include the following:

- maneuvering capabilities through high-level METHODS;
- possession and management of sensors and sensor contacts;
- possession and management of weapons;

- fully automatic animation on a map;
- a Distributed Interactive Simulation (DIS 1.3) interface, including the ability to represent platforms controlled by other simulations using DIS;
- integrated experimental design construction;

All of these features are designed with emphasis on ease-of-use.

At the soul of a platform is its ability to model navigation in its world. The platform needs to know where it is and where it is going, whether it is moving in an arc or a straight line, how fast it is moving, and what time it will complete its current move. Navigation capabilities also include instantaneous changes in speed or course, and some simple management of fuel. All navigation is done in three-dimensional coordinates.

Layered on top of the navigation capabilities are all of the animation routines. Whenever any movement command is given to the platform, the command is intercepted and a two-dimensional graphics command is added. These animation commands are implemented in SimGraphics [9] commands. Each platform has an iconic representation on the geographic situation display. As the platform is commanded to move around, the icon is moved as well. As three-dimensional SimGraphics METHODs become mature, we may change the Foundation to three-dimensional animation.

Interfacing with the movement and graphics METHODs is accomplished using Foundation maneuvers. A maneuver is simply a named series of navigation movement commands. A good example of a maneuver is a `RaceTrack`. By specifying the leg length, radius, and speed of the racetrack, we get a specific racetrack pattern which would be appropriate for a specific platform type. The Foundation maintains an accessible database of racetracks, zig-zags, transits, turns, etc., each with its own name. A platform can execute any maneuver at any time by using the maneuver name.

There is a list of collections of maneuvers which we call paths. A platform can call up a specific path and follow it, executing each maneuver in sequence. Paths can be terminated at any time, the platform can switch to a new path at any time, or it can switch to a new path at the conclusion of the current path. Finally, if no new path is specified at the conclusion of a path, the platform can repeat the path (the default) or stop at its conclusion.

Above-the-line METHODs can invent and store maneuvers and paths as required, modify existing maneuver parameters, or collecting novel combinations of maneuvers into new paths. Finally, a platform can possess above-the-line reactions to sensor input and internal ques.

The system manager level of a platform is where the essential systems of the platform, the sensors and weapons, are managed. Each platform has a suite of sensors mounted on it, as well as an inventory of weapons. When a sensor makes a detection, it informs the host platform that there has been a detection, and passes whatever sensor information is indicated. The platform system manager can also deploy weapons by name and target, but the conditions under which a weapon is deployed vary from platform to platform. Hence, the decision process for weapon deployment is over-the-line, but the capability to deploy a weapon is provided by the Foundation.

On top of all of these capabilities are a simple damage model, the DIS interface (described below), and a database interface which allows the platform to collect all of the information about its systems, movement performance, fuel, icon, and initial planned movement.

3.2 Sensors

Each `SensorObj` possesses a collection of objects we call the virtual sensors. For each sensor in the simulation and each target platform that that sensor might possibly detect, there exists a `VirtualSensorObj`. The virtual sensor computes the time that the target platform it is attached to will enter its detection range. It will also determine the time of the closest point of approach (CPA), as well as the time that the target platform exits the detection range. These events (entrance, CPA, and exit) are rescheduled each time that the target platform or the platform on which the sensor is mounted changes its navigation characteristics. These calculations, although challenging to develop for each possible movement situation, are much more efficient than repeatedly checking each platform pair to determine if there should be a detection.

Many simulations which model platforms and sensors use a time-stepped METHOD for calculating the times of the entry, CPA, and exit. Each δt time units the simulation stops to determine

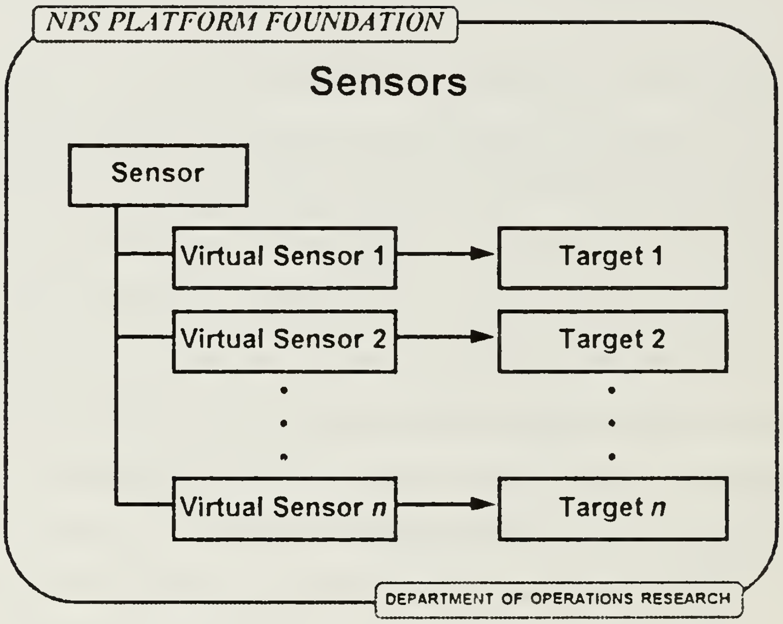

Figure 3: Virtual sensors connecting sensors to targets

the distance between each platform/sensor pair. Suppose that we have N platforms, and we observe a total of t time slices and M maneuvers. Assume $M << t$. A time-stepped approach would execute the distance calculation $O(tN^2)$ times, while our virtual sensor architecture would do $O(MN)$ recalculations of the event times. We clearly enjoy computational superiority to any time-stepped METHOD.

Among the advantages of the object-oriented virtual sensor architecture, we can have a different range determined for each sensor-platform pair, determine velocity-specific, altitude-specific, cross-section-specific, or aspect-specific detection ranges. Each sensor, begin a collection of virtual sensors, maintains a list of platforms along with the true movement status of each. Any error model which the modeler chooses to employ can be implemented in an above-the-line virtual sensor.

Sensors are graphically represented as range rings. The ring is animated by changing color when detection events occur. Unfortunately, each virtual sensor in a sensor may have a different range, but the graphical representation of the sensor allows only a single range. For this reason, we may see an icon inside a range ring without seeing the ring change color. This is because the virtual sensor connecting the sensor to the target platform has a range different than the ring radius displayed.

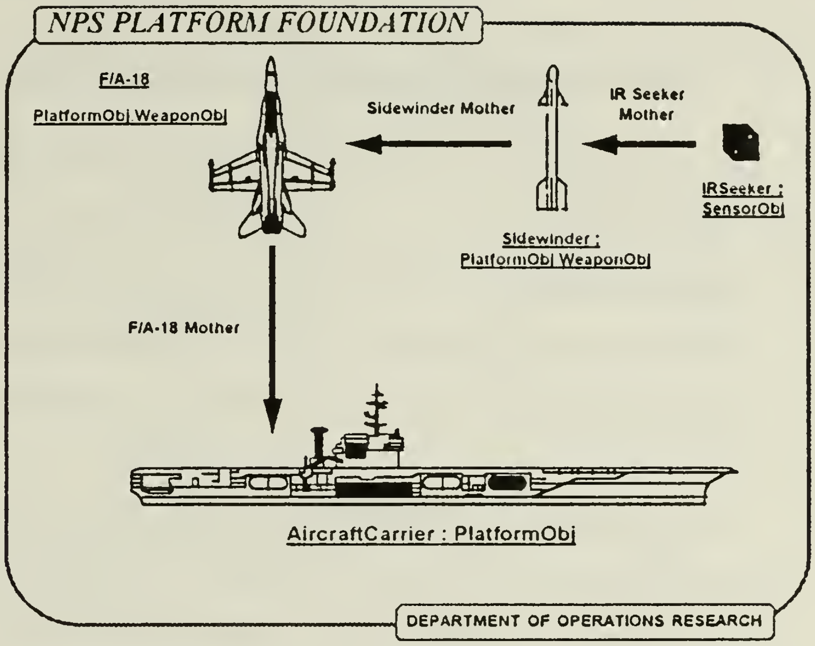

Figure 4: The relationship between weapons, sensors, and platforms in an AAW/CAP application of the Foundation.

3.3 Weapons

Each platform is capable of housing an inventory of weapons. Weapons are special platforms – they have sensors, they move, they are animated, have fuel capacities, and can themselves be detected, damaged or destroyed. Weapons have specially built maneuvers based not on geography, but on the position and movement of the designated target. Special animation features of the generic Foundation platform allow animation of damage and destruction caused by weapon impacts. Weapon effects, while above-the-line, are implemented through the provided platform damage models.

Weapons are deployed through METHODs of the system manager on each host platform. Though somewhat counterintuitive, any offboard sensor platform such as a tethered submarine decoy, a sonobuoy, a chaff canister, or a rescue beacon package can be thought of as a weapon without the ability to directly inflict damage.

Figure 3 shows a possible arrangement of sensors, weapons, and platforms.

4 Distributed Foundation Simulation

Each `PlatformObj` has the capability to produce packets containing its movement, sensor, and weapon activity at any time. Each packet, called a Protocol Data Unit (PDU), is formated in accordance with the Distributed Interactive Simulation (DIS) protocol, [8]. When a Foundation model is put together, the facility required to multicast PDUs is included. In addition, each platform in a Foundation model is capable of receiving and implementing remote direction in the form of DIS PDUs.

When a platform is instantiated, it can be designated as a remote or local platform. If local, it acts based on its own logic, sensors, and tactics, and emits PDUs at specified intervals and instances when its state dramatically changes. If the platform is remote, then it is assumed that another simulation is running which will direct the platform using PDUs. This second simulation may be another Foundation-based simulation or use some other simulation methodology. Thus, Platform Foundation models can run within a distributed simulation model.

In distributed mode, there exist two restrictions required to make simulation smooth and accurate:

1. Every Foundation-based simulation must have each of the platforms in the entire simulation identified and labeled as local or remote. *No remote platforms may be instantiated during the simulation.*

2. There needs to be a time synchronization mechanism in each component simulation which ensures that the model isn't too far away from running in scaled real time. The scale parameter must be the same for each component simulation.

Our methodology used to make DIS PDUs transfer and become implementable can be understood if taken in two parts. The first part of the process is establishing the capability to actually transfer packets across the Defense Simulation Internet. Within each LAN running a component simulation, one workstation must be designated as a DIS bridge to other LANs. This workstation will receive PDUs from other component simulations on other LANs and multicast the PDUs to each workstation

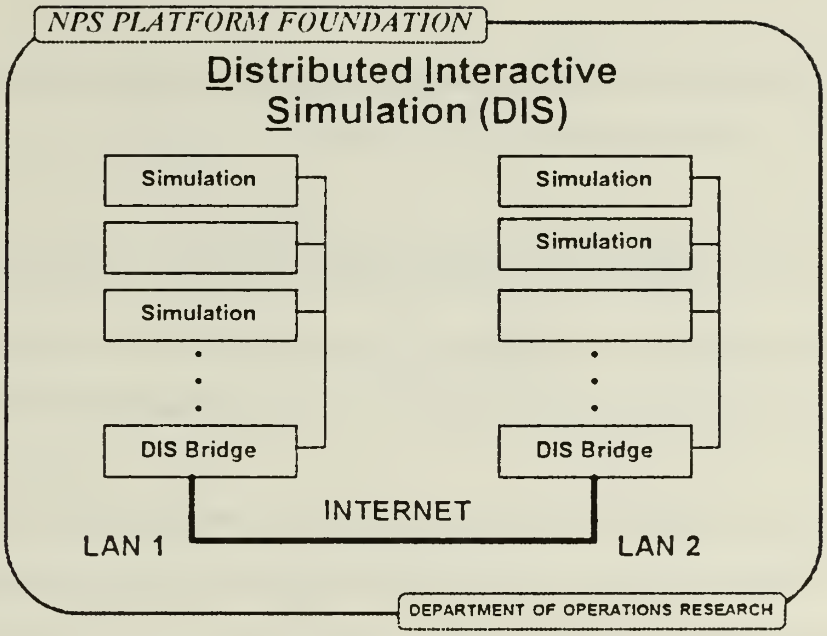

Figure 5: DIS bridges connecting two LANs running component simulations in a distributed simulation model.

in the LAN it serves. Each component simulation running within the LAN also multicasts its PDUs, and one of the recipients is the DIS bridge workstation. Upon receipt of a locally generated PDU, the DIS bridge broadcasts the PDU to other DIS bridges on other LANs. DIS bridge connections are called DIS sockets, and must be point-to-point – TCP/IP and internet do not allow interLAN multicasts. See figure 5.

The second part of the remote direction process involves the handling of PDUs by remote platforms. If we have N component simulations in the distributed simulation, we will have exactly one local version of each platform, as well as N-1 remote copies. The local version will issue PDUs at fixed intervals, as well as times of detection or weapon deployment. The N-1 remote versions of the platform will all receive the PDUs sent out by the local version. Each remote platform will use the PDU timestamp and movement values to deadreckon to a new location and begin movement as dictated by the PDU. Weapons deployments indicated by PDUs are done as close to the PDU timestamp as possible.

The timescale restrictions placed on all component simulations make time synchronization of the component simulations acceptable without having to implement sophisticated synchronization schemes. Our timing approach is essentially conservative, which is completely appropriate given the graphical nature of Foundation applications.

5 Some Example Foundation Applications

In this section, we briefly describe some applications which we plan to build using the Platform Foundation objects. Each model developed has been used to answer a specific study question.

5.1 What is the Tactical Value of adding GPS to Sonobuoys?

This question was addressed in a Master's thesis by LT Jon Baca, USN [4]. In this simulation, we modeled current P-3 Orion tactics used to prosecute a submarine. We built a special sonobuoy `WeaponObj` which was deployed according to representative doctrinal patterns. Due to the drifting of the sonobuoy in the ocean and the unknown influence of the wind on the flight of the sonobuoy from the P-3 to the ocean surface, the exact location of each sonobuoy is not known with much accuracy. The P-3 must continuously fly directly over a sonobuoy to accurrately determine its position, a process called *marking on top*. Thus, our sonobuoys are platforms whose motion is determined by wind and ocean currents, and whose position estimate possessed by the P-3 deteriorates in accuracy over time. Inaccuracy in sonobuoy perceived position leads to inaccurate submarine fixing, and may greatly extends the length of a prosecution. Longer prosecutions lead to lower success probabilities for the P-3.

By adding global positioning system (GPS) capability to a sonobuoy, the position of each sonobuoy is known with near certainty. The P-3 is free to chase the submarine instead of its own sonobuoys. This new efficiency loosens the constraints under which the current tactics were developed. In this simulation, we can statistically and visually compare old and new tactics for submarine prosecution, and we may demonstrate the tactical value of adding GPS to sonobuoys.

5.2 What are Efficient Countermeasure Tactics Against LOFAR Accoustic Detection?

Low Frequency Active Ranging (LOFAR) accoustic detectors are considered a high priority threat to submarines operating in littoral areas. These detectors are long-range, early warning assets used by defensive forces to sense the presence of submarines near shores. The submarine community is currently considering several countermeasures to combat LOFAR, including decoys, sector jamming, and aspect management.

In the simulation developed by LT John Kelley, USN [6], the LOFAR detection mechanisms are sensors which have above-the-line virtual sensor designs. These virtual sensors work only when an accoustic transponder *pings*, and determine whether a detection is made based on a detailed shallow water accoustic model. The LOFAR sensors are mounted on slow-moving or stationary genaric platforms.

One or more submarine platforms are employed in the simulation. These submarines use below-the-line METHODs to employ the countermeasures involving motion. They are stocked with decoys and offboard jammers, all modeled as `WeaponObjs`, and have specialized deployment METHODs for these. The submarine can detect the pinging of the LOFAR transponder.

In addition to the usual display of the sensors and platforms in the scenario, there exists a second geographic display which shows the tactical picture as seen by the LOFAR detection center. The simulation has been used to determine the effectiveness of a set of countermeasure tactics – the LOFAR perception display next to the true geographic display will provide excellent opportunities to evaluate countermeasure tactics. This is done by allowing players to allocate ASW assets, (platforms themselves) to attack the submarines in a video game.

5.3 What is an Effective Deployment Strategy for Coast Guard Cutters?

From a set of origination points, a sequence of drug smuggling craft embark for a set of drop points in U. S. coastal waters. U. S. Coast Guard, U. S. military, and foreign government interdiction assets are deployed to detect, board, and interdict these smugglers. The smugglers are intermixed with a huge amount of civilian and commercial shipping and air traffic, as well as legal and illegal land border crossings. In this simulation, detection and interdiction assets are modeled as platforms with special METHODs for

- sorting out suspicious shipping, air traffic, and ground targets from the rest of the traffic;
- coordination of detection, interception, boarding, and seizure opportunities;
- execution of standard patroling patterns;

Command and control objects are used to manage the assets and to build schedules which obey crew endurance, ship endurance, and maintainence constraints.

The benign and smuggler traffic are generic platforms generated in a somewhat random fashion. This model gives decision makers opportunities for examining schedules and schedule constraints, force structures, and prosecution policies. This simulation is being used to test polices based on route-asset indexing, [2].

5.4 What is an Effective EA-6 Jamming Policy?

In this simulation, there are many, many generic platforms corresponding to radar sites which support surface-to-air missles. These radar sites have a `SensorObj` corresponding to each radar mode, communicate through a command and control network, and house a complement of missles, and don't move. A set of simple air platforms are driven through the lay-down of radar sights on the map, and excite the command and control network. They follow a path which takes them over a target area, where they drop bomb `WeaponObjs`. These aircraft are collectively called the *protected entity*

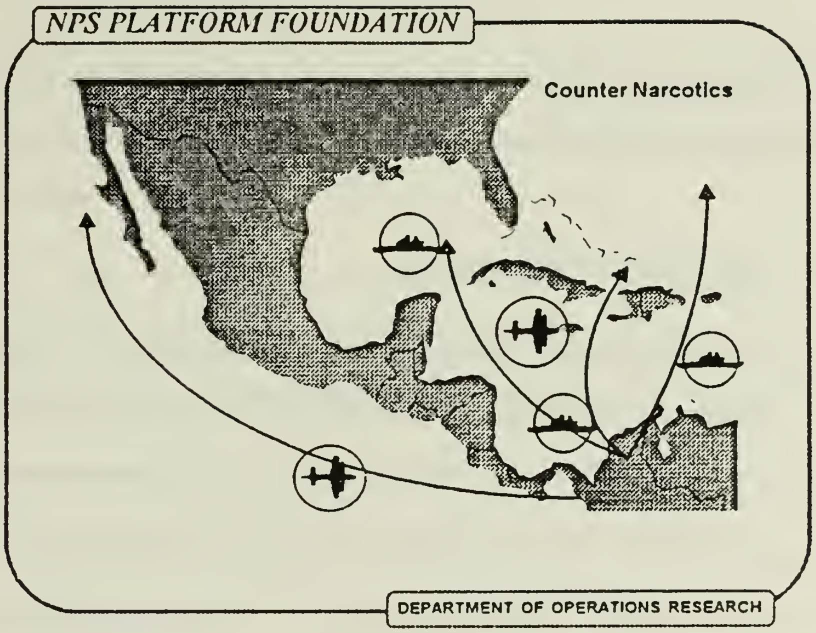

Figure 6: Coast Guard Model uses Platforms to Model Smugglers and interdiction Assets. The animation is used solely for validation, testing, and publicity. The model is used to test indexing policies which provide near-optimal allocation of assets to smuggling routes.

Finally, there are a small number of tactical airborne jamming platforms, EA-6's, which also fly through the area but on paths different than the protected entity. Their mission is to choose a small number of the radar sites to jam, and to jam these radars according to a jamming schedule. The problem of developing the schedule is greatly complicated by the capability of the radars to share information over the command and control network. The schedule must address immediate needs for protection as well as information denial.

6 Conclusion

The NPS Platform Foundation is a useful tool for building simulations focused on platforms, sensors, weapons, and tactics. The Foundation is useful to military operations analysts in the same way that commercially available communications system simulation packages are useful to communications engineers. The Foundation is built in the object-oriented simulation language MODSIM, and has been designed to be extendable to specific engagement simulations easily.

The Foundation objects provide strong support for navigation, sensor-platform interactions, weapon management, damage modeling, visualization, and distributed simulation. By inserting specific tactics, an analyst can build a sophisticated engagement simulation quickly and easily.

Acknowledgements

The NPS Platform Foundation was originally just a class project in course OA4333, Advanced Simulation Methodology. It has grown to its current capabilities because all of those involved in this course enthusiastically continued their contributions beyond what was required in the course. In addition to the author of this work, major contributions to the Foundation were made by:

- GRAPHICS: LT Robert Schultz, USN, LT Michael Dorko, USN, and LT Wu, Hsi-Hsien, Taiwanese Navy;
- SENSORS: LT Jon Baca, USN, LT Jack Noel, USN;
- DIS: LT Donald Brutzman, USN, LT David Schiffman, USN, and LT Robert Murphy, USN;
- MANEUVERING: LT James Dettbarn, USN, LT John Long, USN, and LCDR Alberto Santos, Brazilian Navy;
- DOCUMENTATION: LT Kenneth Unger, USN.

References

[1] Bailey, M. P. 1993. *Object Oriented Simulation Pictures (OOSPics) for Design and Testing.* Technical Report NPS-OR-93-01, Department of Operations Research, Naval Postgraduate School, Monterey, California.

[2] Bailey, M. P. and Kevin G. Glazebrook. 1994. *Indexing Policies for Drug Interdiction.* Technical Report under development, Department of Operations Research, Naval Postgraduate School, Monterey, California.

[3] Bailey, M. P. 1994. *The Prowler IADS Performance Evaluation Tool (PIPE)* . Technical Report under development, Department of Operations Research, Naval Postgraduate School, Monterey, California.

[4] Baca, Jon L. 1993. *An Evaluation of Global Positioning System Enhancements to Sonobuoys in a Simulated P-3 Anti-Submarine Warfare Prosecution.* Master's of Science Thesis, Department of Operations Research, Naval Postgraduate School, Monterey, California.

[5] Booch, Grady. 1991. *Object-Oriented Design.* Redwood City, California: Benjamin Cummings.

[6] Kelley, John. 1994. *A Game to Test the Effectiveness of Countermeasures against Low Frequency Active Ranging Sonar.* Master's of Science Thesis, Department of Operations Research, Naval Postgraduate School, Monterey, California.

[7] *MODSIM II Reference Manual.* 1993. LaJolla: CACI Products Division.

[8] *Proposed IEEE Standard, Draft – Standard for Information Technology – Protocols for Distributed Interactive Simulation Applications, Version 2.0.* 1993. Institute for Simulation and Training, Orlando, Florida.

[9] *SimGraphics Reference Manual.* 1993. LaJolla: CACI Products Division.

INITIAL DISTRIBUTION LIST

1. Research Office (Code 08) 1
   Naval Postgraduate School
   Monterey, CA 93943-5000

2. Dudley Knox Library (Code 52) 2
   Naval Postgraduate School
   Monterey, CA 93943-5002

3. Defense Technical Information Center 2
   Cameron Station
   Alexandria, VA 22314

4. Department of Operations Research (Code OR) 1
   Naval Postgraduate School
   Monterey, CA 93943-5000

5. Prof. Michael P. Bailey (Code OR/Ba) 50
   Naval Postgraduate School
   Monterey, CA 93943-5000